Disclaimer: *The work carried out in this project has not been re-used from any another course project at Indian Institute of Technology Kanpur or at any other institute, or any other project that might have been done elsewhere by us.*

Content Based Music Genre Classification using Temporal and Spectral Features

Debanjan Chatterjee
20111016

Mayank Bansal
20111032

Gaurav Tank
20111407

1 Problem Description

Almost everyone listens to music. It propagates feelings and thoughts between people. Today's music corpus is very heterogeneous. Everybody has their specific music taste mainly because of the diversity of composers and musicians out there. Many industrial streaming services like Spotify, Amazon Music, Pandora use state-of-the-art (SOTA) algorithms to find similarities and patterns in the music tracks, and based on that, it categorizes the tracks into different classes (genres) of the tracks. Moreover, based on the genres one listens to, the algorithm can recommend related songs. As music listeners, we are motivated to get insight into some of the existing music classification techniques and try out experiments using different machine learning algorithms and see if we can improve any of these existing techniques.

2 Literature Review

Besides the content-based music genre classification, other techniques exists as well such as:-

Collaborative filtering: From [7], this approach makes a prediction about the taste of a user with the help of part of the community that shares the same (or similar) taste (thus called collaborative). An important assumption is made that if a user, say A, listens to the same songs of a genre as the users B, C, D and E, then A would also prefer the songs of other genres listened to by B, C, D and E.

The drawback of this technique would be, for the collaborative filtering approach to work, there must be a large set of users (i.e. the community) and user data.

Knowledge-based: This approach (from [1]) draws in user interests and feedback at regular periods. This technique is used mainly when the other two (content-based and collaborative filtering) cannot be applied.

This approach depends on user feedback (an example is the like and dislike option provided by Spotify for each song). Thus, inadequate feedback or unable to obtain feedback regularly hinders the working of the approach. This can be considered as a drawback.

3 Novelty

Deep learning based methods are fairly popular when it comes to problems such as music genre classification. However, music (audio) data has a nice sequential structure. The order of data is important, and data across the time-stamps should not be treated as independent. Recurrent neural networks address this issue as they are networks with loops in them, allowing past information to persist. LSTM networks are a special kind of RNN, capable of learning long-term dependencies. In the project we have proposed a hierarchical LSTM-based model for the multi-class classification problem.

We adopted the hierarchical LSTM architecture from [9] with some modification mentioned below:

- We proposed two different kind of approaches hard prediction and soft predictions.
- We are using sequence length = 256.
- We added Chroma-STFT, Spectral Centroid and Spectral Contrast features.

4 Proposed Methodology

4.1 Feature Extraction

4.1.1 Features for Audio Signals

In computers, audio track is represented as digital signal. Digital signal is discretized representation of the analog signal and it is sampled at some sample rate. Audio track is either recorded using microphone or synthesized within computer itself.

Digital audio signal feature extraction is done emphasizing these fine grained features like frequency, amplitude, phase, pitch, timbre, amplitude envelope, power of signal and tonal features.

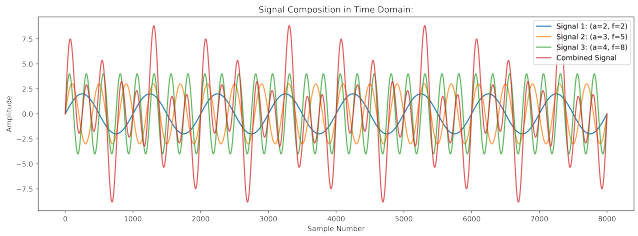

Figure 1: Representations of Signals in the Time Domain sampled at Rate=2000Hz

Audio Signals are represented into two domains namely (1) Time Domain and (2) Frequency Domain. Both empirically and theoretically it is known that complex signals are easy to decompose in the frequency domain. Figure 1 shows the a signal in the time domain and Figure 2 shows the same signal in the frequency domain.

The time domain to frequency domain conversion can be done using algorithms like Fast Fourier Transform (FFT). Which is calculated using equation (1) below.

$$y(f) = \int_{-\infty}^{\infty} y(t) e^{-i2\pi ft}\, dt \tag{1}$$

Where $e^{-i2\pi ft}$ is the complex signal with frequency = f and $y(t)$ is our signal. In Discrete Fourier Transform (DFT), we can use the dot product to find the similarity between our signal and the complex signal with frequency f. And we can do this for all frequencies between f_{min} to f_{max} to find the transformed signal.

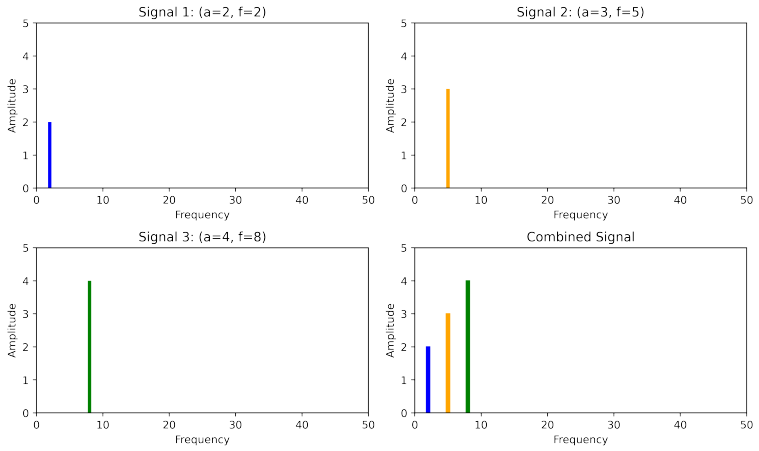

Figure 2: Representations of Signals in the Freq Domain sampled at Rate=2000Hz

4.1.2 Mel-Spectrogram

Fourier transform of the signal gives the spectrum for whole signal at once. So, it will give global features. However, we are interested to find the non-stationarities in the signals as we are using recurrent models like LSTM. To achieve this, we can use something like windowing method. and perform (Short Time Fourier Transform) STFT on each window which convolves over our signal with stride equal to hop-length. This will give something like spectrum.

Frequency perception in human is non-linear in nature and spectrum captures the frequencies in the linear scale. Mel-Spectogram is varient of spectogram which captures frequency on Mel-Scale which is similar to logarithmic scale.

Figure 3 and 4 shows an example of signal in time domain and its Mel-Spectrogram respectively.

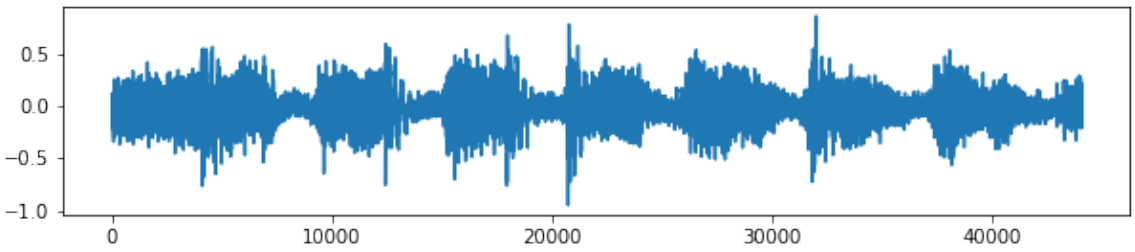

Figure 3: Signal in Time Domain

Mel-Spectogram facilitates almost good features but it is high dimensional. Some methods like (Kong et al.)[4] use CNN (Convolutional Neural Network) based approaches and it use this as feature. Since we are using LSTM based approach, we can use denser features like MFCC, Chroma-STFT, Spectral Centroid and Spectral Contrast which is described in the next section.

4.1.3 Audio Features for Our Network

We use 22050 Hz sampled, 30 second long audio file. We are taking window size = 2048 samples and hop-length = 512 samples. So, we will have approx 1290 timestamps which is calculated using formula below:-

$$\text{No of Windows (Timestamps)} = \frac{(\text{Track Length} * \text{Sample Rate}) - \text{Hop Length}}{\text{Hop Length}}$$

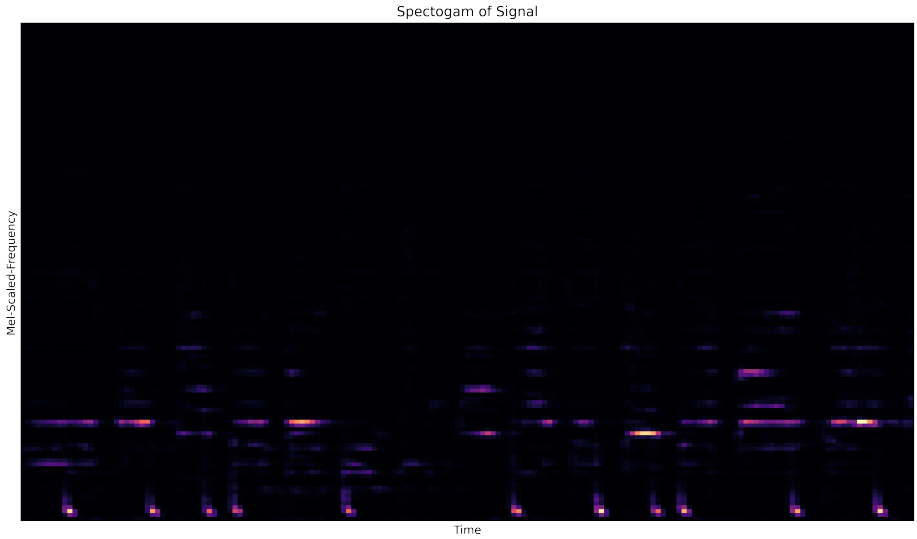

Figure 4: Signal in Mel-Spectogram

Here, each window acts as a timestamp for our LSTM network. Since LSTM has restriction on the sequence length, we fix the sequence length = 256. To do this we split the tracks into 5 segments such that each segment has 256 timestamps. Note that here number of timestamps is the same as number of different windows on which we are finding the features.

Our dataset had initially 1000 songs but after enforcing the sequence length = 256, we are kind of augmenting the dataset such that it will have total of 5000 segments each of sequence length = 256.

Now for each timestamp, we calculate following features:

Features	Dimensionality
MFCC	20
Chroma-STFT	12
Spectral Centroid	1
Spectral Contrast	7

The MFCC Features captures the details about the envelope in IDFT of the Log Power Spectrum which is extracts similar features from Mel-Spectogram but represented in densed way. We are using 20, Chroma-STFT features finds the Intensity for different tones of the current window. There are 12 tones namely (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) So, Per window its dimensionality is 12. Spectral Centroid captures the average frequency weighted by intensity at that frequency. Per window its dimensionality is 1. And Spectral Contrast is an alternative to MFCC features which works well to do genre classification task as described in paper [5]. Per window its dimensionality is 7. So combining all above features, total we have 20 (MFCCs) + 12 (Chroma-STFT) + 1 (Spectral Centroid) + 7 (Spectral Contrast) = 40 features.

So, our dataset has total 5000 examples (after augmentation), each example has sequence length of 256 timestamps and each timestamp has dimensionality = 40.

4.2 LSTM networks as Multi-class classifier

To achieve multi-task classification, we have used a LSTM-based model. LSTM[3] is good technique to use for music genre classification as it remembers the past result of the cell in the recurrent

layer and classify music in a better and efficient way. The equations for the LSTM modules are given as follows:-

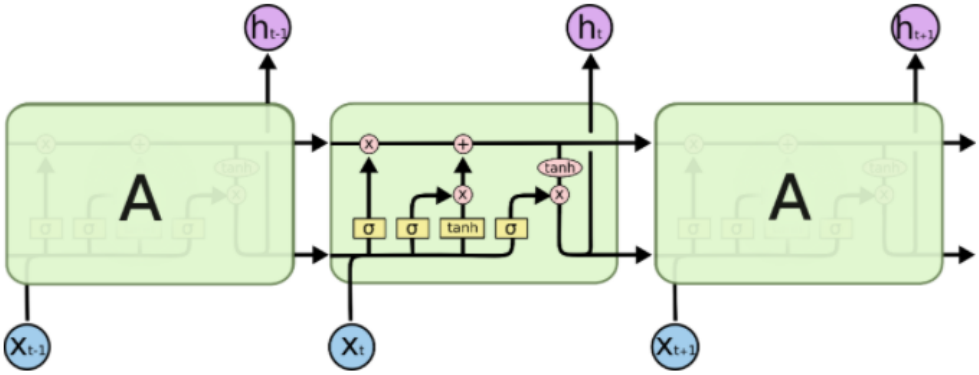

Figure 5: The repeating module in an LSTM contains four interacting layers [2]

$$u_t = \tanh\left(W_{xu} * x_t + W_{hu} * h_{t-1} + b_u\right) \text{: update equation}$$
$$i_t = \sigma\left(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i\right) \text{ : input gate equation}$$
$$f_t = \sigma\left(W_{xf} + x_t + W_{hf} * h_{t-1} + b_f\right) \text{ : forget gate equation}$$
$$o_t = \sigma\left(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o\right) \text{ output gate equation}$$
$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \text{ : cell state}$$
$$h_t = \tanh c_t \odot o_t \text{ : cell output}$$
$$\text{output class} = \sigma\left(h_t * W_{outpara}\right)$$

where $W_{xu}, W_{xi}, W_{xf}, W_{xo}$ and $W_{hu}, W_{hi}, W_{hf}, W_{ho}, W_{outpara}$ are weights, and b_u, b_i, b_f, b_o are biases to be computed during training. h_t is the output of a neuron at time t. $\odot$ denotes point-wise multiplication. σ denotes a sigmoid function and $tanh$ represents the tanh function. The input x_t is the MFCC parameters at time t. output class is the classification output.

Instead of using a single LSTM network to perform a 10-class classification task, we use a divide and conquer approach, by using a hierarchical tree-based 7 LSTM network architecture. The following figure shows the proposed hierarchical LSTM architecture.

The functionality of each of the LSTM networks is given as follows:

- **LSTM 1:** It classifies between strong (hip-hop, metal, pop, rock, reggae) and mild (jazz, disco, country, classic, blues) genres of music
- **LSTM 2a:** It classifies between sub-strong 1 (hip-hop, metal, and rock) and sub-strong 2 (pop and reggae)
- **LSTM 2b:** It classifies between sub-mild 1 (disco and country) and sub-mild 2 (jazz, classic, and blues)
- **LSTM 3a:** It classifies between hip-hop, metal and rock
- **LSTM 3b:** It classifies between pop and reggae.
- **LSTM 3c:** It classifies between disco and country.
- **LSTM 3d:** It classifies between jazz, classic, and blues.

This hierarchical architecture, helps to tackle the multi-class classification problem, by a divide and conquer based approach, where each LSTM in the tree, is trained using samples of the relevant classes. The idea for the hierarchical LSTM model was adopted from [9].

The architecture of the individual LSTMs is given as follows:

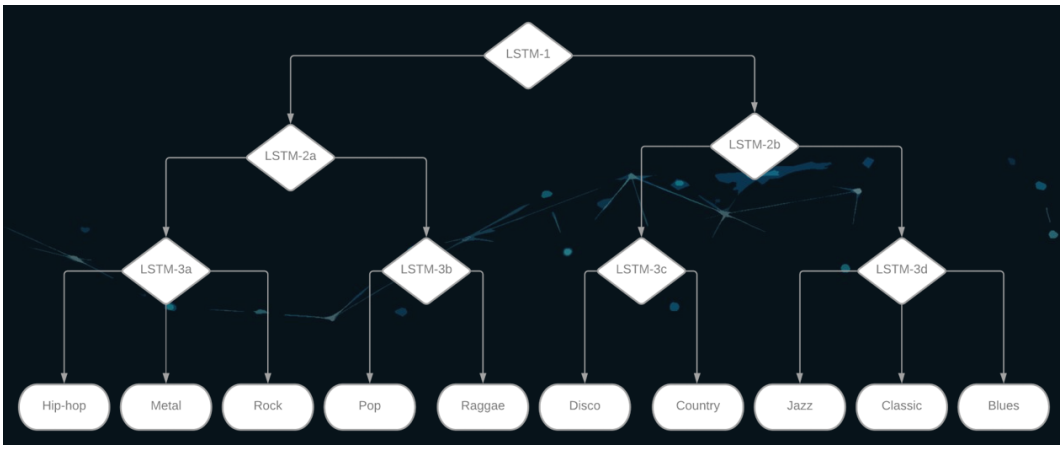

Figure 6: The proposed hierarchical LSTM architecture

Input layer	Total 40 features (MFCC, Chroma STFT, Spectral Centroid, Spectral Contrast)
Hidden Layer I	64 LSTM units
Hidden Layer II	32 LSTM units
Output Layer	Number of softmax units depends on which the number of fan-out results.

5 Experimental Results

The code as well as the results are available at our Github repository: Audio-genre-classification

5.1 Dataset Used

The dataset used in the project is the GTZAN dataset[10] available at the MARYSAS website. This dataset was originally used in[10]. The dataset is made up of about 1000 audio tracks each of which is 30 seconds long. There are about 10 genres, each covering up 100 tracks for each genre. All tracks has 22050 Hz sampling rate, mono channel with 16-bit sample audio files in .wav format.

The audio files were gathered using different sources like CD's, radio, microphone recordings etc. to represent a variety of recording conditions.

5.2 Libraries Used

- Librosa[6] - Librosa is a package used in python for music/audio analysis. We used Librosa to extract features from the GTZAN dataset.
- TensorFlow [8] - TensorFlow is a very popular open source platform widely used in Machine Learning. It comprises of numerous tools, libraries etc. which helps in development purposes. We used TensorFlow to build LSTM (Long Short Term Memory) model for our project.
- Matplotlib - Matplotlib is a python library that is used for building static as well as dynamic graphs. We have used Matplotlib to create plots.
- Numpy - Numpy is used for carrying out mathematical and scientific computations.

5.3 Train-Validation-Test Split

The dataset was split into 75 percent for train, 15 percent for validation and 10 percent for test sets, while applying the stratified property.The stratified property was applied as, it is desirable to split the dataset into train and test sets in a way that preserves the same proportions of examples in each class as observed in the original dataset, thus avoiding class imbalance. The total number of samples for

LSTM1, LSTM2a, LSTM2b, LSTM3a, LSTM3b, LSTM3c, LSTM3d are 5000, 2500, 2500, 1500, 1500, 1000, 1000, 1500 respectively.

5.4 Results and Analysis

We have used two prediction methods:

Hard Prediction

- In this, we perform segmentation on the original track so that each segment will have 256 timestamps after feature extraction.
- Predict the class for each segment.
- Select label with highest frequency.
- For all segments, select the path with majority predicted label.
- Repeat for all internal nodes until leaf nodes are not predicted.
- Return the predicted label.

Soft Prediction

- In this, we perform segmentation on the original track so that each segment will have 256 timestamps after feature extraction.
- Predict the class for each segment.
- For each segments, choose path independently.
- Do this for all internal nodes until leaf nodes are not predicted.
- Return all predicted labels with the frequencies.

Figure 7, Figure 8 and Figure 9 shows the Iteration vs, Cost plot for the 7 LSTMs. Figure 10 depicts the confusion matrix hard prediction, soft-prediction(Top-1) and soft-prediction(Top-2) respectively. As we can see that using hard prediction method, accuracy is as good as mentioned in [9]. Using soft prediction method, it improved from 53 percent to 63 percent, and using top-2 predictions it is improved to 80 percent. In the case of top-2 predictions if any of the two genres (classes) with the highest predicted probabilities matches with the ground truth label, the prediction is deemed as correct. The reason for the accuracy of top-2 predictions being much higher than the top-1 prediction can be attributed to the fact that some genres like pop and reggae, are hard to distinguish even by an avid music listener.

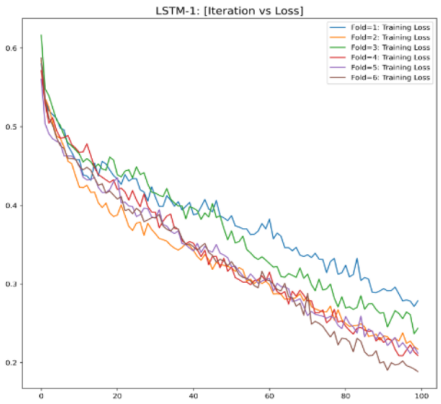

Figure 7: Iteration vs. Cost plot for LSTM-1

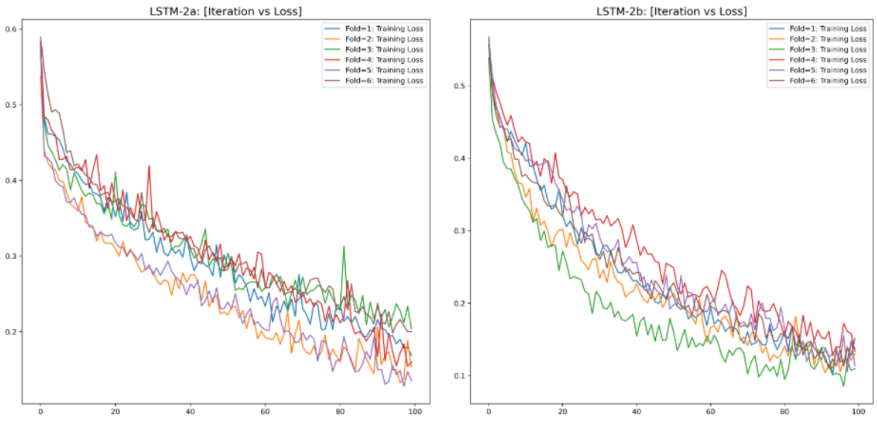

Figure 8: Iteration vs. Cost plot for LSTM-2a (left) and LSTM-2b (right)

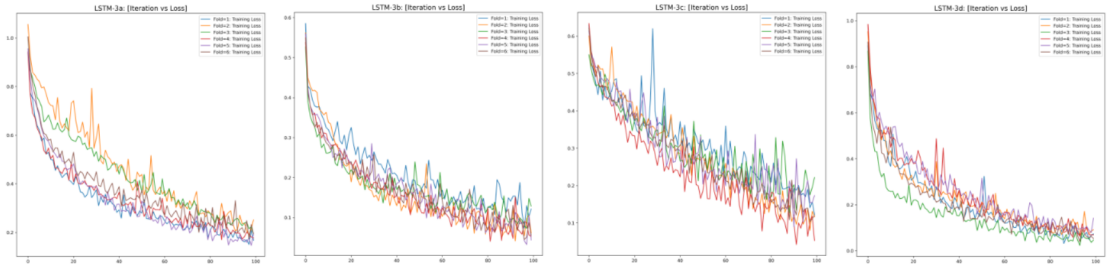

Figure 9: Iteration vs. Cost plot for (from left right) LSTM-3a, LSTM-3b, LSTM-3c and LSTM-3d

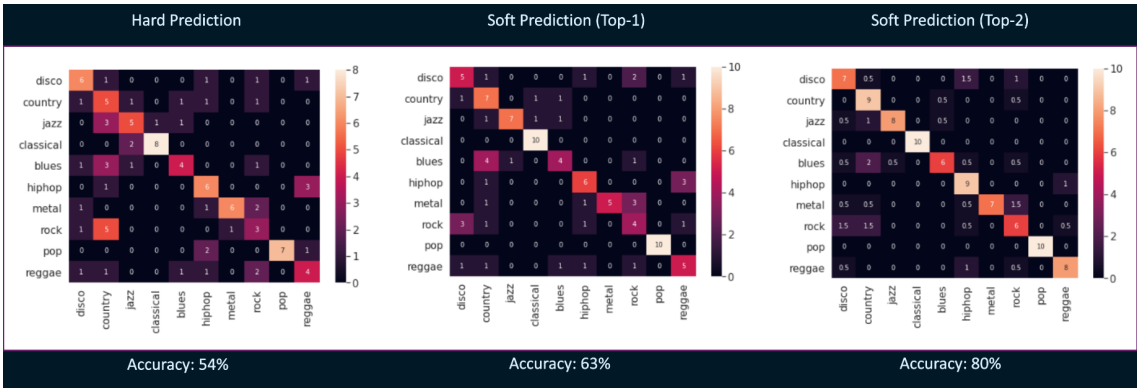

Figure 10: Results evaluation for Hard prediction, Soft prediction (Top-1) and Soft prediction (Top-2)

6 Discussion

The content based music classification approach is being used in the industry widely. As there are numerous machine learning techniques that work well on extracting pattern, trends or other useful information from a large dataset, thus these techniques are suitable for performing music analysis.

Several companies are using music classification to segment their database according to genre and are either using it to recommend songs to their users like done by Spotify, Youtube Music etc. or even purely as a product like Shazam.

7 Possible Future work

A shortcoming of content based approach is that it fails to account for the cultural as well as contextual meaning of the item (in our case song) and only look for similarity between the items. Also, with the content based approach, there is little room left for surprise as out of the box recommendations are not done with this approach which can help the user to explore his/her taste.

We plan to integrate our content based approach with the collaborative filtering method so as to overcome the above two shortcomings. This will also provide us with the added benefit of using the information collected from the community.

We also plan to introduce a feedback mechanism which would allow us to draw in feedback from the users and thus improving the current approach even further.

References

[1] Charu C. Aggarwal. *Knowledge-Based Recommender Systems*, pages 167–197. Springer International Publishing, Cham, 2016.

[2] Colah. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, Aug 2015. Accessed on 2020-11-22.

[3] Douglas Eck and Juergen Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In *Proceedings of the 12th IEEE workshop on neural networks for signal processing*, pages 747–756. IEEE, 2002.

[4] Qiuqiang Kong, Xiaohui Feng, and Yanxiong Li. Music genre classification using convolutional neural network. In *Proc. of Int. Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[5] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Jung-Mau Su. Automatic music genre classification using modulation spectral contrast feature. In *2007 IEEE International Conference on Multimedia and Expo*, pages 204–207. IEEE, 2007.

[6] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[7] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[8] Open source. Tensorflow.

[9] Chun Pui Tang, Ka Long Chui, Ying Kin Yu, Zhiliang Zeng, and Kin Hong Wong. Music genre classification using a hierarchical long short term memory (lstm) model. In *Third International Workshop on Pattern Recognition*, volume 10828, page 108281B. International Society for Optics and Photonics, 2018.

[10] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293 – 302, 08 2002.